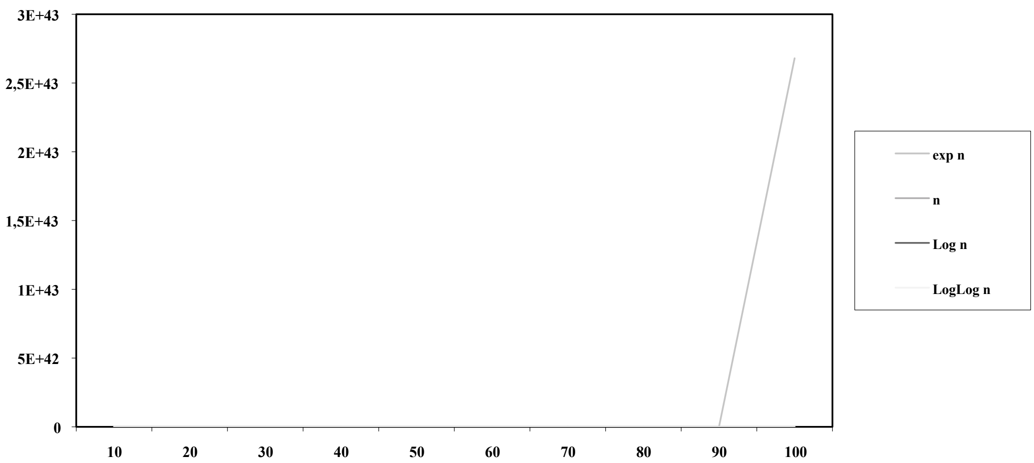
# Designing ultra-fast algorithms

Jean-Louis Roch
Ensimag, INP, Grenoble University
INRIA / LIG Moais

- Multiple Access (SDMA / FDMA / CDMA ) provides concurrent access to memory in constant time :

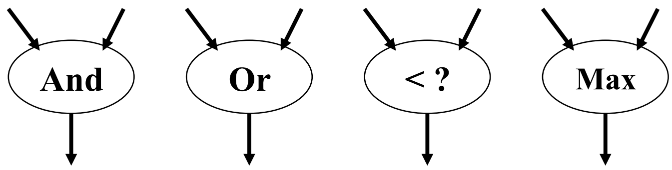
CRCW : Concurrent Read Concurrent Write

# Algoritmic costs

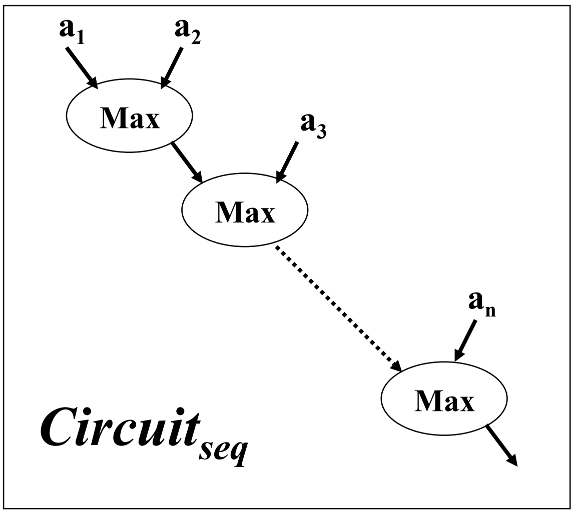


# Computing the maximum

- Designing the fastest circuit to compute the maximum
- Input : n elements $a_i$ of an ordered set $<$
  Output : the maximum element
- Available gates :




# Basic serial circuit

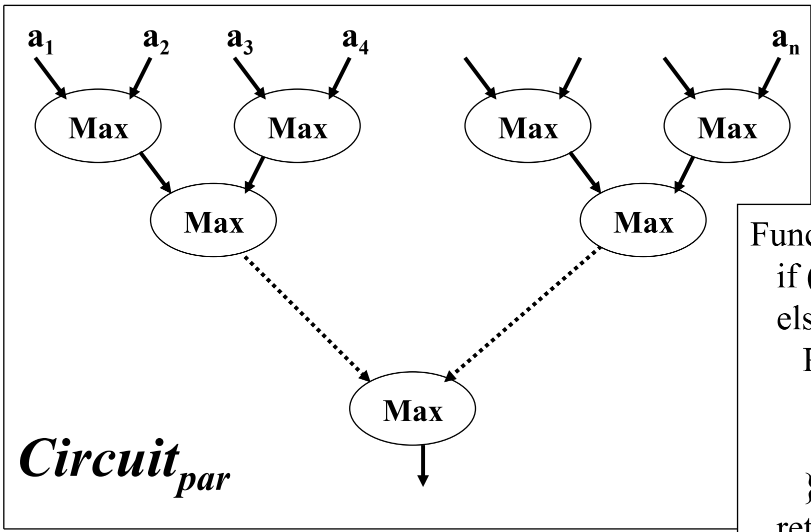



```
res := a1 ;
For i :=2 .. n do
  res := Max ( res, an ) ;
Return res;
```

D=Depth= n W = Work = n

(NB Work and depth are in number of Compare gates)

# Faster with Parallelism



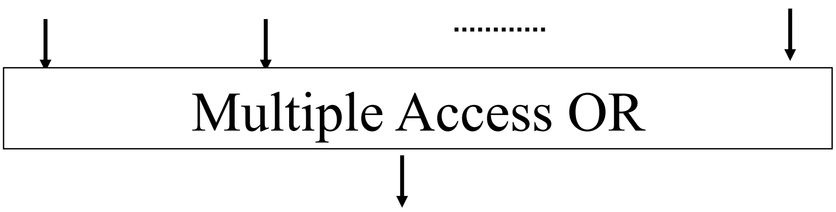

```
Function max2(ai .. ak ) {
  if (i == k) return ai ;
  else {
    PARALLEL {
      rl = max2(ai .. a(i+k)/2 ) ;
      rh = max2(a(i+k)/2+1 .. ak) ;
    }
  return Max( rl, rh ) ;
}
```

$D = \log_2 n$ $W = n$

# May Multiple Access help ?

- Taking benefit of multiple access :

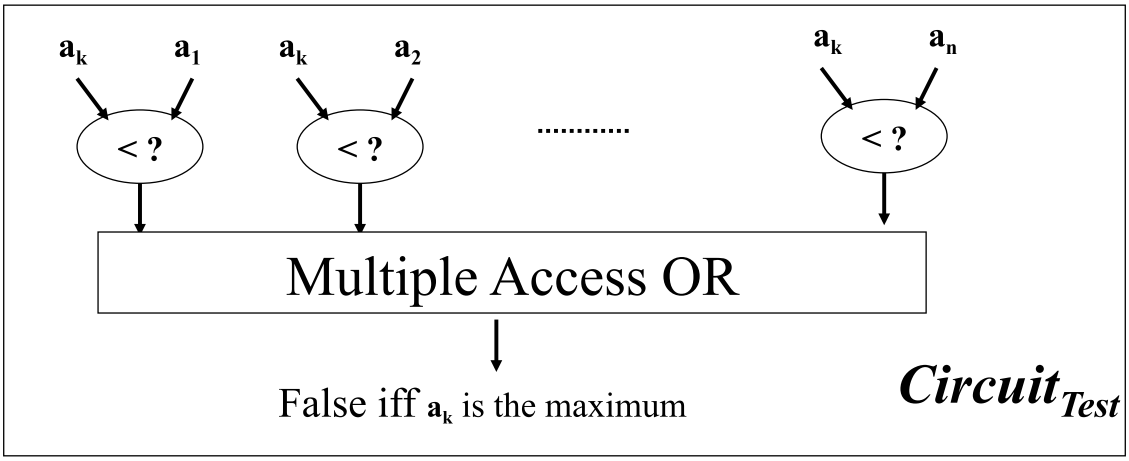
logical or of n bits in constant time



# Ultrafast algorithm for testing the maximum

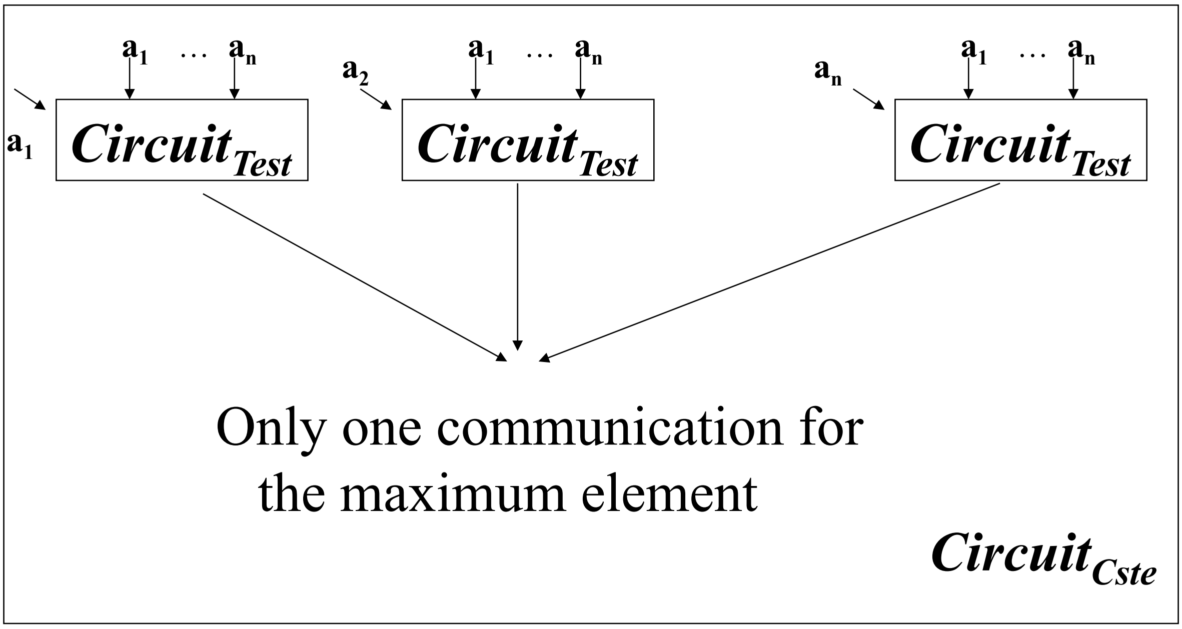
$\mathbf{a}_k = \mathrm{Max}(\mathbf{a}_1 .. \mathbf{a}_n) \Leftrightarrow \mathbf{a}_k \geq \mathbf{a}_i$ for $i \neq k \Leftrightarrow \mathbf{AND}_{i \neq k}(\mathbf{a}_k \geq \mathbf{a}_i)$

$\Leftrightarrow \mathbf{NOT}\ (\mathbf{OR}_{i \neq k}(\mathbf{a}_k < \mathbf{a}_i))$



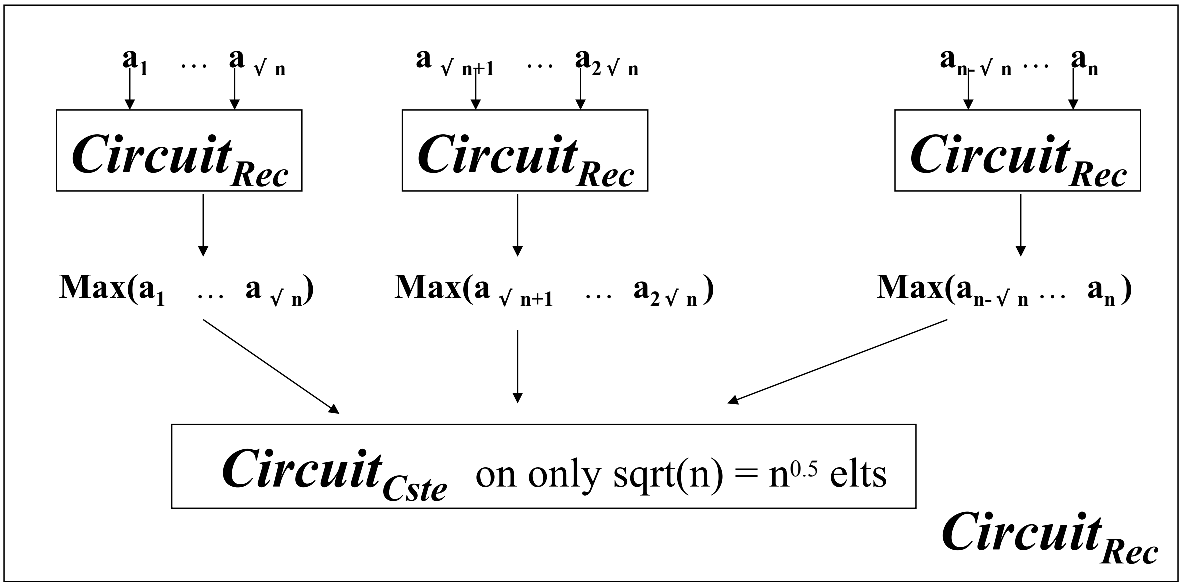

D= 1 ☺ W= n

# Application: computing the maximum




**D= 1** ☺ **W= n²** ☹

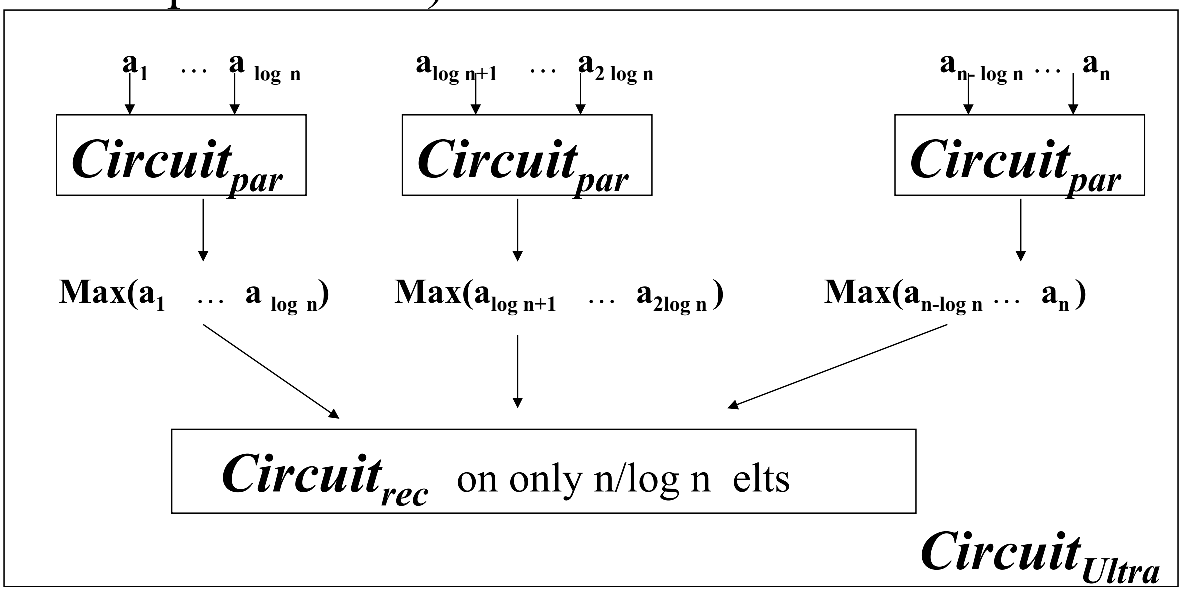
# A recursive ultrafast parallel algo




- $D(n) = D(n^{0.5}) + 1 = \log\log n$
- $W(n) = n^{0.5}.\#procs(n^{0.5})n + n = n \log\log n$

# Optimizing the number of units

- Take benefit of the parallel algorithm to minimize the number of units (could be the sequential one)



# Conclusion : an ultrafast algorithm

- Final algorithm : D = depth= 3.loglog n ☺
  W = work= 1.n + O(n loglogn / log n) ☺
- Technique used : « cascading »
  mixing 3 algorithms to obtain an ultrafast one !

- Fundamental technique for parallel algorithms design and in software engineering too.
- Both theoretical and practical issues.

- Exercises:
  - Circuit for Merge on CREW PRAM in D=loglog n and n ops
  - D&C Program for Prefix with depth=O(log n) (not $\log^2 n$!) (in Kaapi or Cilk or TBB or …)